Internal Version (Early Draft)

Smart Navigation Strategies for Virtual Landscapes

Henrik BUCHHOLZ, Johannes BOHNET, Jürgen DÖLLNER

1 Introduction

Navigation is a key factor for user acceptance of virtual 3D landscapes. Existing geovirtual environments (GeoVEs) frequently suffer from the lack of a proper handling and prevention of confusing or disorientating situations. As FUHRMANN & MACEACHREN (2001) point out, “core problems for users of these desktop GeoVEs are to navigate through, and remain oriented in, the display space and to relate that display space to the geographic space it depicts.” This paper proposes smart navigation strategies, which overcome these problems and give additional features to landscape designers for user guidance:

- Smart navigation strategies interpret user interaction regarding the current view specification, i.e., the parameters of the virtual camera, and determine if the user is about to get into confusing or disorienting situations in an anticipatory way.
- They guide the user away from situations where usual navigation behavior tends to fail.
- They always indicate to the user when the guidance mechanism is operating, so that the user understands the behavior of the smart navigation strategy.
- They allow for constraining the camera according to data quality and the emphasis given to certain parts of the virtual landscape.

With smart navigation strategies we aim to achieve a higher user acceptance for virtual landscape applications. Particularly, smart navigation strategies facilitate the use of visualization applications for inexperienced users without the need for a specific training.

2 Motivation

2.1 User Acceptance

In the field of landscape visualization the usability of an application depends on navigation techniques as primary user control. Whether a user accepts an application is to a considerable degree determined by the very first experiences with it. If the user has to spend a significant amount of time on learning how to navigate, or if the user loses control in certain situations, an application will quickly be rejected regardless of the possible advantages it offers. Particularly, this applies to applications that are to be used by a broad audience with different skill levels or in situations in which users have to be convinced to continue within a few minutes of trial. Examples of those applications include virtual landscapes used in tourism information systems, urban planning visualizations for public participation, or public transport information systems.

2.2 Problems of Existing Applications

Most 3D visualization applications provide a number of navigation techniques such as the virtual trackball or airplane navigation. User interaction events such as mouse events or keyboard events are directly translated to movements of the virtual camera. This provides an intuitive control mechanism as long as the user behaves as the application designer intended. In practice, however, this assumption about the user's behavior leads to several problems:

- The user navigates into a view specification that does not provide sufficient information for the user to keep oriented inside the virtual landscape, e.g., if only the sky is visible.
- In the case of multiple navigation techniques or automatic camera animations the behavior of a newly activated navigation technique may not be defined for the current view specification, e.g., if a camera path ends with a distant overview of the whole landscape from a high altitude and the user switches to the walker navigation mode.

These problems along with camera-environment collisions are commonly handled by simply blocking the navigation. This, however, has a frustrating effect on the user, particularly because the reason for the blocking might not be obvious.

3 Related Work

3.1 Navigation Techniques

A variety of different established navigation techniques have been developed in the past. Many of them are suitable for certain application scenarios in landscape architecture and public participation. They can be classified into egocentric and exocentric frame of reference (HAND, 1997) and distinguished according to the task they are suited for (TAN ET AL, 2001).

With the walker navigation-technique the user explores a virtual environment from the point of view of a virtual avatar, which can walk in four directions and rotate the gaze around two axes using the mouse. The walker navigation-technique is an essential feature of visualizations for public participation, because they allow users for experiencing a virtual landscape from a natural perspective.

For large areas, additional navigation techniques are needed to overcome the inherently slow movement of the walker navigation-technique and to provide surveys of an area. One example is the airplane navigation-technique, which controls a virtual flying vehicle. The user manipulates the speed of the forward/backward movement and the rotation around the world's up-vector. In addition, the flying vehicle allows for changing the height and for tilting the view direction. The airplane technique is frequently used in geovirtual environments.

A useful extension to these techniques is the combination with a click-and-fly technique, which allows a directed flight to a target point selected by a mouse-click on the screen (MACKINLAY ET AL, 1990). Navigation can be further improved by a landmark-selector technique, which calculates on-demand camera paths from the current position to certain pre-defined viewpoints (HELBING & STROTHOTTE, 2000; SALOMON ET AL, 2003).

All techniques described above are well suited to present results of a landscape planning process to the public. For the planning process itself, however, other navigation techniques are preferable that enable more rapid movement and fast changes of the viewing scale.

One example is the combination of trackball navigation-technique, zoom navigation-technique, and move-focus navigation-technique. Using the trackball, the user moves the camera on the surface of a virtual sphere. The zoom technique allows for controlling the distance of the camera to a focus point. This focus point is determined by shooting a ray from the camera position towards the view direction. With the move-focus technique the user switches from the current focus point to a new one. For this, the move-focus technique rotates the camera so that the selected point moves to the center of the screen. There is a large number of other useful techniques (DARKEN & SIBERT, 1993; HAND, 1997; IGARASHI ET AL, 1998; TAN ET AL, 2001).

3.2 Constrained Navigation

For the problem of constrained and guided navigation different approaches have been developed. In the river analogy proposed by GALYEAN (1995) the user is guided by a predefined path and controls the gaze direction along with slight deviations away from the path. HANSON & WERNERT (1997) extended this concept to guide manifolds. Using 2D input devices the user moves on a designer-provided surface. The remaining degrees of freedom are controlled by pre-defined guide fields according to the camera position. In WERNERT & HANSON (1999) this method has been extended and applied to collaborative virtual environments. The attentive camera (HUGHES & LEWIS, 2002) addresses the problem of guiding the view direction without distracting the user from the intended walk direction. KISS & NIJHOLT (2003) presented a system that alters the view direction based on the terrain slope and some objects of interest around the camera position. Each object of interest is suggested to the user by shortly focusing it. The system also ensures that the user is always aware of obstacles that prevent the user from moving.

4 Smart Navigation Strategies

Smart navigation strategies split common navigation techniques into two steps. First, the mapping from user interaction events to camera movements takes place. Second, the intended movement is checked against several constraints and modified, if necessary. A schematic illustration of this process is given in Fig. 1. The *Navigation Mapper* determines the intended camera movement from the user input. This information is passed to the *Constraint Checker*, which validates the intended camera movement regarding the current view specification and a set of constraints. If necessary, the camera movement is modified. Finally, the camera movement alters the view specification.

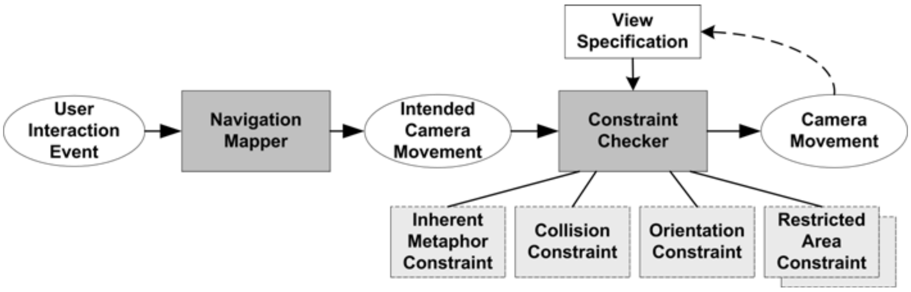

Fig. 1: Schematic illustration of the navigation process.

The constraints include:

- the *collision constraint*, which avoids collisions of the camera with landscape objects;
- the *orientation constraint*, which guarantees that the view contains a significantly high value of orientation supporting information;
- the *restricted area constraints*, which are user defined constraints for the given landscape;
- the *inherent metaphor constraint*, which guarantees that the camera behaves as specified by the navigation metaphor, e.g., the walker navigation strategy keeps the camera on a certain height above the ground.

The constraint checker uses the intended camera movement to interpolate future camera positions and orientations. By this, necessary modifications of the camera movement can be regarded in an anticipatory way. Additionally, the concept of smart navigation strategies allows for switches between different smart navigation strategies. Therefore virtual landscape applications can make use of several navigation strategies applying the optimal navigation strategy depending on the current user task.

We will describe the beneficial use of smart navigation strategies in the context of common scenarios in the visualization of virtual landscapes. The strategies are able to identify a variety of situations and to react on them in a user supporting way. In the following, we characterize these situations.

4.1 Anticipatory Collision Detection

The constraint checker of each smart navigation strategy regards the user’s intended camera movement and therefore considers potential conflicts with constraints at an early stage. Collisions of the camera with objects in the landscape are avoided in an anticipatory way.

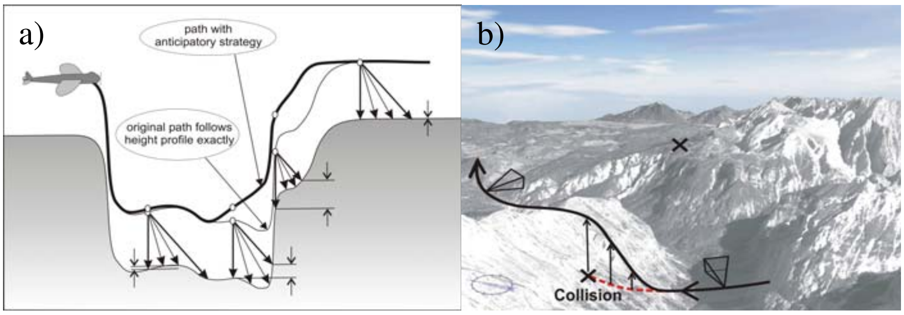
a)
path with anticipatory strategy
original path follows height profile exactly
b)
Collision

Fig. 2: Examples of anticipatory collision prevention.
a) Anticipatory flight path using airplane navigation strategy.
b) Anticipatory sideward movement using trackball navigation strategy.

Fig. 2a illustrates the anticipatory collision strategy for the airplane navigation. Usually, the airplane navigation technique would keep the flight height above ground and the camera would follow the height profile exactly. The anticipatory strategy measures the terrain heights in front of the airplane using a small number of test rays, ranging from 90°, looking down to the terrain, and 45°, tilted up towards the estimated flying path. Next, the maximum of all measured terrain heights is calculated, and the height difference between the maximum and the terrain height straight below the camera is determined. Finally, the camera height is increased by this difference. This way, the camera is lifted in advance when approaching an elevation and is prevented from making jerky movements in the case of approaching a sharp increase.

For the trackball navigation, a collision constraint observes whether the camera is approaching an elevation that the user does not see as the camera is moving sideward and the elevation is still outside the view frustum as illustrated in Fig. 2b. The constraint increases the camera height dependent on the distance of camera position and the elevation, so that the camera gets over the obstacle smoothly.

4.2 Orientation Information

To define the quality of a view specification we introduce the *orientation value,* which measures the amount of visible information that helps users to orient themselves within the geovirtual environment. For example, we can calculate the orientation value by measuring the portion of the screen that is covered with relevant parts of the scene (e.g., terrain surface, buildings, vegetation). An enhanced implementation could also consider the visibility of landmark objects for determining the orientation value.

The orientation constraint checks the orientation value of a view specification and corrects the camera movement in a way that the user orientation is always maintained.

The simplest method to maintain a high orientation value is to block the proposed movement by using the previous view specification. Blocking the movement, however, usually leads to disturbing effects. We propose an improved method, which identifies the user’s intention and guides the user away from disorienting views based on the intention and outweighing the degraded orientation with corrections in orthogonal movement degrees.

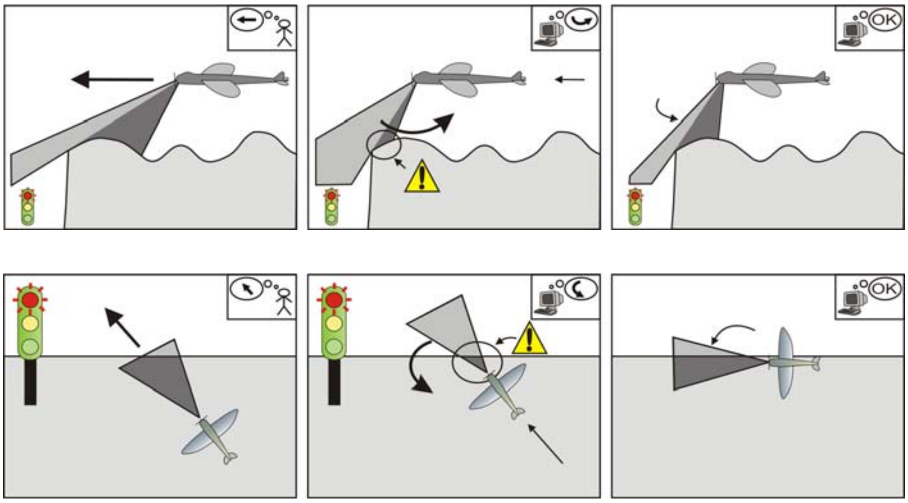

Fig. 3: Orientation constraint for keeping high orientation values.
top row: Constraint checker tilts down the view direction.
bottom row: Constraint checker turns left.

Fig. 3 illustrates the orientation strategy of the airplane navigation. In two typical scenarios the user is about to navigate into a critical situation:

- The user is flying forward beyond the terrain border. The orientation constraint temporarily tilts down the view direction up to a maximum tilting angle.
- If no more tilting is possible, the strategy rotates the flight direction parallel to the terrain border.

4.3 Restricted Areas

Designers can mark certain areas of the virtual landscape as restricted by defining *restricted area constraints*. The virtual camera is only allowed to approach these areas up to a certain minimum distance. Restricted areas are handled similarly to terrain borders (see section 4.2). When the user is approaching a restricted area, the constraint gets activated and prevents the user from entering the area. The guidance strategy either guides the user away from the restricted area as in the case of a terrain border, or it increases the camera height to pass the area holding the minimum distance.

Restricted areas can be used to adapt a visualization to data quality of different parts of a virtual landscape. On the one hand, user movement can be kept away from areas of low data quality. The user just gets an overview of that area without noticing the poor quality. On the other hand, the designer can constrain the user view to a certain area with high data quality to emphasis on these areas.

4.4 Initialization of Navigation Strategies

After switches between different smart navigation strategies or after an automatic camera path animation the newly activated navigation strategy is confronted with an arbitrary camera position and orientation. The strategy may not be able to work properly if the view specification is not compatible with the underlying metaphor of the new strategy. The constraint checker detects this problem and invokes a short animation leading to a suitable view specification. During this animation, however, the user will not lose control. The user can interrupt the animation at any time by using a different navigation strategy.

Fig. 4 illustrates a switch from the walker navigation to the trackball navigation. The walker navigation allows the user to look into an arbitrary direction, e.g., the sky. After the switch to the trackball navigation it is likely that no valid focus point is defined, i.e., sending a centered ray into the landscape does not allow for obtaining an intersection point to be defined as focus point. Therefore, the orientation checker automatically tilts down the view direction until the ray intersection produces a valid focus point. Now the trackball navigation can be used in the usual way.

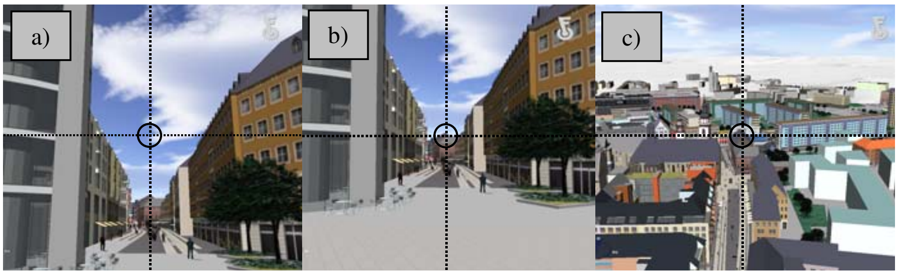

Fig. 4: Initialization process for the trackball navigation-strategy.
a) Uninitialized Trackball. b) Initialized Trackball. c) Trackball in use.

By providing the automatic initialization methods for each navigation technique we aim to motivate users to switch more between different navigation strategies if it is appropriate. If users have problems with their first attempts to switch between different strategies, they tend to keep using only one, possibly inappropriate, navigation strategy, just to ensure familiar behavior of the camera control. Automatic initialization methods let multiple smart navigation strategies appear as a single combined navigation tool with a compatible set of control alternatives.

4.5 Cues

An important issue for the usability of navigation controls is the user's ability to understand their behavior permanently. As long as the camera is controlled directly by mapping user input to camera movement or camera speed, e.g., when using walker navigation, the navigation behavior is straightforward to understand. Each kind of intervention into the direct user control, however, is usually not understood by inexperienced users. Therefore, an application has to inform the user when navigation is affected. This occurs in three situations:

- The user has – consciously or unconsciously – activated an automatic camera animation, i.e., to restore a previously stored camera position.
- The user has switched to a new navigation strategy that has to be initialized first.
- The constraint checker intervenes to ensure that all active constraints are maintained, e.g., if the user approaches a restricted area.

For each of these situations an application must provide a cue to inform the user, that an intervention takes place and why. Since audio devices are not always connected, such cues should usually be visual. In the case of camera animations of known time-duration, it is also useful to inform the user about the progress state of the animation, e.g., by showing a small progress bar.

Although the cues are important for user-acceptance, they should not appear too dominant within an application. A user must be able to concentrate on a certain task instead of the navigation itself. So, visual cues should be as concise as possible. Traffic signs provide a good example for visual cues: While concentrating on traffic car drivers must notice the signs and understand them quickly. Analogously, in virtual landscapes a flashing stop sign can indicate, when a user approaches a restricted area and the constraint checker intervenes.

5 Conclusions & Outlook

Considering domain-specific information effectively improves user navigation in geovirtual environments. This way, we can implement assisting, automated navigation controls that guide the user through the virtual environment while giving a high degree of control to the user. Smart navigations play an important role in applications targeting on a broad audience with inexperienced users, such as in supporting public participation by interactive visualizations in urban and landscape planning processes. In particular, our concept of smart navigation strategies provides means to improve the usability of interactive landscape visualizations by preventing users from "getting lost" in the geovirtual environment and by informing users about the navigation strategy's behavior.

The concepts of smart navigation have been developed within the scope of the Lenne3D project, which is developing an interactive landscape planning and real-time visualization system. As a future work, we plan to evaluate and optimize smart navigations strategies for interactive public participation processes in the scope of landscape planning and urban reshaping. In addition, we plan to investigate how to take advantage of object semantics to further automated navigation.

Acknowledgement

We would like to thank the German environmental foundation Deutsche Bundesstiftung Umwelt for supporting our work within the Lenné-3D research project (www.lenne3d.de) as well as our project collaborators. We are also grateful to IMK automotive GmbH, Chemnitz, for providing us the city model data of the city of Chemnitz (Fig. 4).

6 References

Formatted: Bullets and Numbering

Darken, R.P. & J.L. Sibert (1993): *A Toolset for Navigation in Virtual Environments.* Proceedings of the ACM User Interface Software and Technology, 157-165.

Fuhrmann, S. & A.M. MacEachren (2001): *Navigation in Desktop Geovirtual Environments: Usability Assessment.* Proceedings 20th ICA/ACI International Cartographic Conference.

Galyean, T.A. *Guided Navigation of Virtual Environments* (1995): Proceedings 1995 Symposium on Interactive 3D Graphics

Hand, C. (1997): *Survey of 3D Interaction Techniques.* Computer Graphics Forum, 16(5), 269-281.

Hanson, A.J. & E.A. Wernert (1997): *Constrained 3D Navigation with 2D Controllers.* Proceedings of Visualization ’97, IEEE Computer Society Press, 175-182.

Helbing, R. & T. Strothotte (2000): *Quick Camera Path Planning for Interactive 3D Environments.* Smart Graphics’00, http://wwwisg.cs.uni-magdeburg.de/~helbing/

Hughes, S. & M. Lewis (2002): *Attentive Interaction Techniques for Searching Virtual Environments.* Human Factors and Ergonomics Society’s 46th Annual Meeting.

Igarashi T., R. Kadobayashi, K. Mase & H. Tanaka (1998): *Path Drawing for 3D Walkthrough.* Proceedings of UIST’98, 173-174.

Kiss, S. & A. Nijholt (2003): *Viewpoint Adaptation during Navigation based on Stimuli from the Virtual Environment.* Proccedings of the 8th International Conference of 3D Web Technology, 19-26.

Mackinlay, J.D., S.K. Card & G.G. Robertson (1990): *Rapid Controlled Movement through a Virtual 3D Workspace.* SIGGRAPH ’90 Proceedings, 171-176.

Paar, P. & J. Rekittke (2005): *Lenné3D - Walk-through Visualization of Planned Landscapes*. Visualization in landscape and environmental planning, 152-162.

Salomon, B., M. Garber, M.C. Lin & D. Manocha (2003): *Interactive Navigation in Complex Environments Using Path Planning.* Proceedings ACM SIGGRAPH Symposium on Interactive 3D Graphics, 41-50.

Tan, D.S., G.G. Robertson & M. Czerwinski (2001): *Exploring 3D Navigation: Combining Speed-Coupled Flying with Orbiting.* CHI 2001 Conference on Human Factors in Computing Systems, 418-425.

Wernert, E.A. & A.J. Hanson (1999): *Framework for Assisted Exploration with Collaboration.* Proceedings of Visualization ’99, IEEE Computer Society Press.